Low Cost Localisation for Agricultural Robotics

Andrew English, David Ball, Patrick Ross, Ben Upcroft, Gordon Wyeth, Peter Corke
Queensland University of Technology
Gardens Point, Queensland, 4000 Australia

Abstract

This paper presents a pose estimation approach that is resilient to typical sensor failure and suitable for low cost agricultural robots. Guiding large agricultural machinery with highly accurate GPS/INS systems has become standard practice, however these systems are inappropriate for smaller, lower-cost robots. Our positioning system estimates pose by fusing data from a low-cost global positioning sensor, low-cost inertial sensors and a new technique for vision-based row tracking. The results first demonstrate that our positioning system will accurately guide a robot to perform a coverage task across a 6 hectare field. The results then demonstrate that our vision-based row tracking algorithm improves the performance of the positioning system despite long periods of precision correction signal dropout and intermittent dropouts of the entire GPS sensor.

1 Introduction

Automating agricultural operations with autonomous robots will help to address the rising demand in world food production. An important component in autonomous robots is accurate and reliable positioning information. Guidance of large manned agricultural vehicles using GPS has become standard practice for many years in farming sectors such as broad-acre cropping, with the first systems commercialised in the late 90's. Accurate systems still typically cost tens of thousands of dollars and are based on dual frequency multi-constellation GNSS receivers with a wireless correction signal to achieve accuracies of several centimetres.

Small, light, low cost robotic farming machinery has several advantages over the current large, heavy and expensive farm machinery, such as reducing soil compaction and erosion and potentially improving productivity. However, before such robotic systems can be made commercially viable, low-cost precise and reliable positioning is needed to replace the high cost GNSS solutions employed today.

The approach described in this paper mimics the behaviour of a human driver following GPS guidance information, who will also simultaneously pay attention to visual cues (such as the crop rows) and self-motion cues while driving. This strategy allows our system to use lower cost sensors and continue to navigate for extended periods of time without one or more of the information sources. Specifically, the positioning system fuses three types of information; external global information from GPS, internal velocity measurements from both an IMU and wheel odometry, as well as local external cues from a vision-based crop row tracking algorithm.

We test our positioning system by using it to guide a robot to cover 6 hectares of a field on a working farm. Additionally, we investigate the performance of the system under GPS correction signal dropouts, and intermittent GPS signals, and show that the inclusion of our new vision-based row tracking method allows improved accuracy under these degraded conditions.

The following section reviews literature in positioning agricultural vehicles. Section 3 describes the individual sensors and the robot test platform and we explain our approach to filtering the various sensor inputs in Section 4. Finally we demonstrate our experimental setup and results in Section 5 and 6 before concluding remarks in Section 7.

2 Literature

The well-established method of positioning in agricultural fields is using GNSS (Global Navigations Satellite Systems), most commonly the GPS system maintained by the United States government. Agricultural receivers routinely achieve sub-metre accuracy using differential correction data, and more recently have been able to achieve accuracies on the order of 2cm with RTK (Real Time Kinematic) corrections sent by a nearby base station. These highly accurate RTK-GPS systems are costly, and so their use in agriculture is usually restricted to very large and efficient manned machinery.

A recent step towards much lower cost precise positioning is the open source software package RTKLIB [Takasu and Yasuda, 2009] which allows online centimetre-level RTK GNSS positioning using raw GPS data gathered from a variety of low-cost GNSS receivers. Researchers have investigated using RTKLIB with low-cost single frequency GPS receivers and RTK corrections to guide various machines including UAVs [Stempfhuber and Buchholz, 2011] manned agricultural vehicles [Jensen et al., 2012; Osterman et al., 2013] and even an agricultural robot [Takai et al., 2011]. These investigations largely found that accuracies in the region of several centimetres were achievable; however the single frequency receivers typically took many minutes to acquire or re-acquire this level of accuracy when either satellite lock or correction information is lost. This is due to the time needed to estimate ionopheric delays by performing ambiguity resolution [Li and Wang, 2012]. Dual frequency GPS receivers can resolve these errors very quickly, however are presently quite expensive.

Another approach often studied for localising agricultural robotic vehicles is to exploit the semi-structured nature of the field environment, by sensing the location of crop rows relative to the vehicle. Strategies include using monocular vision [Tillet and Hague, 2002; Søgaard and Olsen, 2003] and stereo vision [Kise and Zhang, 2008], as well as scanning lasers [Biber et al., 2010]. These methods typically identify crop rows based on their colour and intensity difference to the soil, or alternatively the difference in height between the crop rows and the soil. A difficulty in relying solely on localising with respect to crop rows is that localisation will often fail in areas of the field where a clear crop row signal cannot be observed. This often occurs in patches of weeds, bare patches, and at the ends of the fields (the headland area) where the row pattern is often overlapping and confusing.

3 System Components

A block diagram of the pose estimation system is shown in Figure 2. Each of the sensors are inputs to a particle filter, which estimates a position and heading. This pose is fed to the vehicle's navigation software which performs GPS waypoint navigation, path planning and obstacle avoidance. The components of positioning system and platform are described briefly in this section.

3.1 Test Platform

The localisation system was tested on a prototype agricultural robotic platform developed for spraying weeds on broad-acre zero-tillage farms. The platform (Figure 1) is based on an electric utility vehicle modified for autonomous operation. Computation is performed on two standard PCs running Ubuntu 12.04 running the open source ROS (Robot Operating System) middleware.

To estimate both linear and angular velocity the robot is fitted with wheel encoders on each of the rear wheels and a low-cost MEMS IMU (UM6 from CH Robotics). The IMU combines rate gyros, accelerometers and magnetometers with an internal Extended Kalman Filter to estimate the vehicle's attitude.

3.2 GNSS Receivers

A high-performance multi-constellation Novatel GNSS with tactical grade IMU and dual antennas was used as a reference positioning system to calculate ground truth. This system can achieve accuracies in the region of 2cm under ideal conditions. RTK corrections were received via the robot's existing 3G network connection from the SmartNet Australia CORS (Continually Operating Reference Station) network. The nearest base station in the network was approximately 35 km away, which is slightly longer than ideal for RTK-GPS corrections.

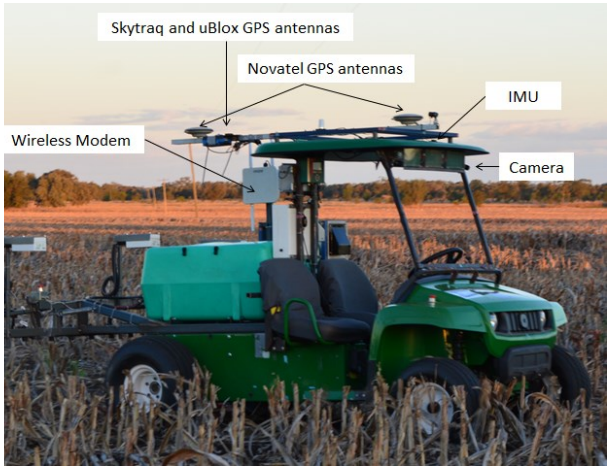

Figure 1: Robotic test platform designed for spraying weeds. Sensors used for calculating robot pose are indicated.

Two different low-cost GNSS receivers were tested; a Skytraq S1315F-RAW, and a uBlox LEA-6T. Both sensors were configured to output raw GPS data for processing by the RTKLIB software package. RTK corrections were applied *only* to the Skytraq, with the uBlox configured in "single" solution mode. This allows a comparison between low-cost RTK-GPS and low-cost uncorrected GPS. Antennas used were the Tallysman TW3100 for the Skytraq, and an ANN-MS for the uBlox.

3.3 Visual Navigation: Crop Row Tracking

An additional navigation input to our system is the use of a novel visual crop row tracking algorithm. This algorithm uses a calibrated forward facing camera (IDS uEye CP) to localise with respect to the crop rows. The row tracking method does not explicitly identify crop rows, but tracks them by estimating the direction and sideways lateral movement of the dominant parallel texture in the image. The image processing pipeline is shown in Figure 3 has the following main steps:

1. Acquire image and pre-process by correcting for lens distortion and downsampling to improve processing speed.
2. Rotate and translate the pre-processed image to remove variations due to the roll and pitch of the vehicle. Vehicle roll and pitch is estimated with a combination of visual horizon detection and IMU data which are combined with a simple Kalman filter. The horizon detection method similar to [Thurrowgood et al. 2009].

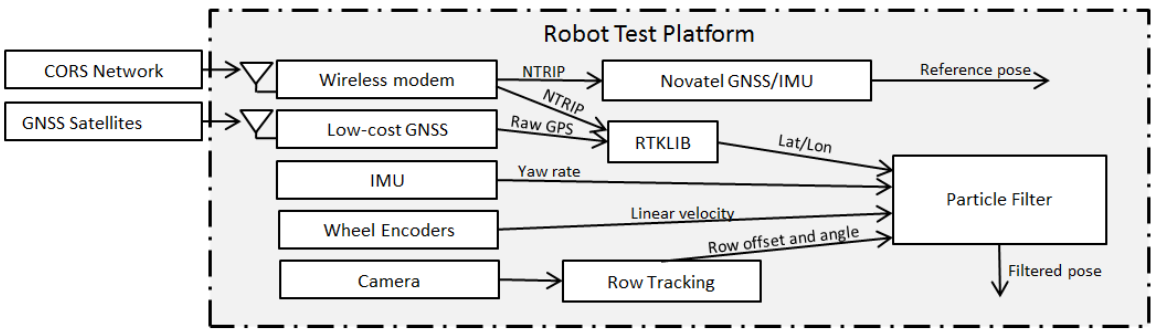

Figure 2: Block diagram of the localisation system components.

3. Warp a region of the stabilized image into an overhead view using planar homography under the assumption that the ground is flat and level.
4. Skew the overhead image to correct for the vehicle heading. Vehicle heading is estimated using a combination of IMU data and estimating the direction of the dominant parallel texture in the overhead image. These two heading measurements are fused with a Kalman filter.
5. Generate a "*crop template*" vector by summing the pixel intensities along the columns of the skewed overhead image.
6. Use this template to estimate lateral motion of the vehicle relative to the crop by comparing the current crop template to the initial crop template using cross correlation.

An advantage of this method of visually tracking crop rows over existing methods is that it abstracts away crop-specific details such as colour, spacing and periodicity and so is virtually calibration free.

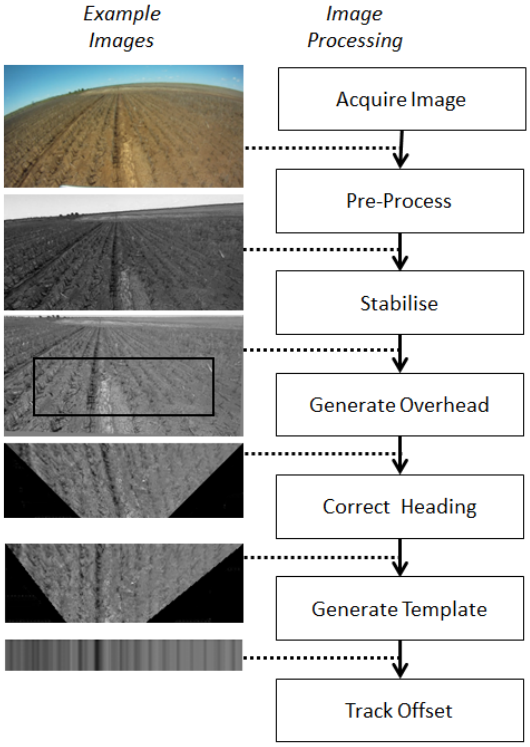

Figure 3: Block diagram of row tracking algorithm.

The measurement produced by the row tracking algorithm and supplied to the particle filter is

$$\boldsymbol{z}_{rows} = \begin{bmatrix} \emptyset_{rows} \\ \tau_{rows} \\ a_{rows} \end{bmatrix}$$

where $\emptyset_{rows}$ is a heading relative to the crop rows, τ_{rows} is a lateral offset since tracking was acquired and a_{rows} is a "available" flag which is 1 when row tracking information is valid, and 0 when continuous row tracking has been lost.

Since our method tracks only the relative offset or crop rows since tracking was acquired, the offset is reset to zero every time tracking is lost or when the measured row heading is greater than +/- 30 degrees. This occurs when turning around at the ends of the field or in bare patches of field.

Although the visual row tracking algorithm can successfully track crop rows with slight curves, in this work we assume the crop rows are planted in straight and parallel rows at a known orientation. This known row geometry allows the crop row measurements to be incorporated directly as an additional observation in the particle filter for estimating vehicle pose. A straight row assumption is reasonable on many broad acre fields where crops are planted in straight parallels passes with a tractor guided by a high accuracy RTK-GPS. Tracking crop rows that are not perfectly straight with these assumptions will therefore introduce additional error in the filter's pose estimate. Future work will look at operating in fields with unknown row geometry.

4 Filtering Sensor Data

Data from each of the sensors is fused with a particle filter. We chose a particle filter for the ease of integrating nonlinear process and measurement models, and the ability to simultaneously track multiple hypotheses.

The state vector for our filter is defined as

$$\boldsymbol{x} = \begin{bmatrix} x_v \\ y_v \\ \theta_v \\ \varepsilon_{\dot{\theta}} \\ \varepsilon_{GPS} \end{bmatrix}$$

where

x_v is the vehicle easting estimated at the GPS, in m
y_v is the vehicle northing estimated at the GPS, in m
θ_v is the heading of the vehicle, in rad
$\varepsilon_{\dot{\theta}}$ drift of the rate gyro, in rad/s
ε_{GPS} is the GPS bias perpendicular to the rows, in m

The state variables x_v and y_v are the estimated coordinates of the vehicle in UTM coordinates at the GPS location G (in Figure 4) which is on the centre line of the vehicle and at a distance l_{AG} ahead of the rear axle G.

The key concept in a particle filter is that the posterior distribution is represented by a set of weighted particles. The filter involves two steps that are performed iteratively, prediction and update. During the prediction step, the values of the state vector in each particle is updated according to the system model, system inputs $\boldsymbol{u}$, along with some random process noise $\boldsymbol{v}$. The update step is performed when new sensor measurements $\boldsymbol{z}$ are available and involves re-weighting the particles with a weight proportion to the probability of the measurement given the particle's current state $p(\boldsymbol{z}|\boldsymbol{x})$. The probability distribution is then updated by sampling particles (with replacement) with a probability proportional to their weight.

4.1 Motion Model

We use a planar motion model for our vehicle and assume no wheel slip. The control inputs to our system are

$$\boldsymbol{u} = \begin{bmatrix} v_A \\ \dot{\theta}_A \end{bmatrix}.$$

The linear velocity v_A is measured by the wheel encoders, and is defined at a point A in the centre of the rear axle (refer to Figure 4). The rotational velocity $\dot{\theta}_A$ is

measured using the IMU. At each time step of interval Δt the vehicle motion is approximated as a small movement forward by $v_A \Delta t$, followed by a rotation about the centre of the rear axel A by an angle of $\dot{\theta}_v \Delta t$. However, since the state vector is centred on the GPS receiver at point G, we must project the point G back onto the axle A, perform the rotation by $\dot{\theta}_v \Delta t$ then project the position back to the GPS centre. This gives us the motion prediction equations:

$$x_{v,k+1} = x_{v,k} + (v_{A,k} + v_v)\Delta t \ cos\,\theta_{v,k} + l_{GA}\left[cos(\theta_{v,k} + \dot{\theta}_{v,k}\Delta t + \varepsilon_\theta) - cos(\theta_{v,k})\right]$$

$$y_{v,k+1} = y_{v,k} + v_{A,k}\Delta t \ sin(\theta_{v,k}) + l_{GA}\left(sin(\theta_{v,k} + \dot{\theta}_{v,k}\Delta t + \varepsilon_\theta) - sin(\theta_{v,k})\right)$$

$$\theta_{v,k+1} = \theta_{v,k+1} + \dot{\theta}_{v,k}\Delta t$$

The state space includes a GPS bias ε_{GPS} since GPS errors are known to be highly time correlated, and hence will appear to "drift" slowly over time. When tracking crop rows for long periods of time, the measured GPS position will slowly drift from the row, causing a conflict between the two measurement methods. This will result in a "jump" towards the GPS position as soon as crop row tracking is lost, which is undesirable. By including this bias in the state vector we can estimate the "drift" in GPS while tracking crop rows to keep the two measurement methods consistent. This is particularly important when using GPS without RTK corrections since GPS errors are comparatively large.

GPS error is often modelled with a combination of various types of white noise and coloured noise [Mao et al., 1999], with a popular method being to model the error as an exponentially autocorrelated random variable [Bar-Shalom, et al, 2004]. Using this model, the update step for our GPS becomes

$$\varepsilon_{GPS,k+1} = e^{-1/\tau}\,\varepsilon_{GPS,k} + v_{GPS}$$

where v_{GPS} is an independent random variable with a normal distribution, and τ is the correlation coefficient. We only track the GPS bias in the direction perpendicular to the crop rows since we have no method of observing the bias in the direction parallel to the rows.

The final state variable to update is the gyroscope drift ε_θ. This is modelled as a random walk

$$\varepsilon_{\theta,k+1} = \varepsilon_{\theta,k} + v_{\dot{\theta}}$$

where $v_{\dot{\theta}}$ is a zero mean white Gaussian random variable.

4.2 Measurement Models

This section describes the measurement model used in our approach. For this work we use two different measurement sources: GPS and vision based crop row tracking. Each of these is applied as they are available, which allows a more robust system that will continue to operate despite dropouts in one of the measurements.

GPS Measurement Model

Since the position of the GPS receiver is estimated directly in the state vector, the measurement equation for GPS observations is simple, however we must also account for the GPS bias ε_{GPS} in the direction perpendicular to the crop rows. This gives us the measurement model

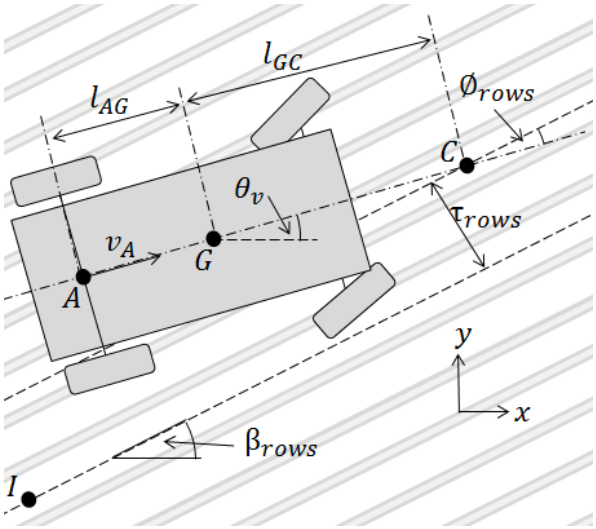

Figure 4: Vehicle model. The GPS measures the vehicle position at point G. Heading and lateral motion of the crop rows (light grey parallel lines) are measured by the camera at point C ahead of the vehicle. The centre of the rear axle lies at point A.

$$\boldsymbol{z}_{GPS} = \begin{bmatrix} x_{GPS} \\ y_{GPS} \end{bmatrix} = \begin{bmatrix} x_v + \varepsilon_{GPS}\, sin\,\beta \\ y_v + \varepsilon_{GPS} cos\,\beta \end{bmatrix} + \boldsymbol{v}_{GPS}\,.$$

We model the measurement noise $\boldsymbol{v}_{GPS}$ as independent Gaussian white noise.

In this work we do not account for the roll and pitch of the vehicle which will affect the position of the GPS antenna, however this effect was found to be minor for the terrain typically seen by the test platform vehicle.

Row Tracking Measurement Model

The visual row tracking estimates the heading and lateral movement of the vehicle with respect to the crop rows at the visual navigation point C (Figure 4) which lies on the ground at a distance l_{GC} ahead of the GPS. The observed crop row heading $\emptyset_{rows}$ is measured relative to the vehicle heading. The observed offset τ_{rows} is the distance the visual navigation point C has travelled perpendicular to the direction of the crop rows since row tracking acquired a lock, at which time point C was located at point I in Figure 4. At the moment the row tracking acquires a lock we store for each particle its hypothesised coordinates of C. That is, for each particle we store

$$I = \begin{bmatrix} x_{rows\_init} \\ y_{rows\_init} \end{bmatrix} = \begin{bmatrix} x_v + l_{GC} \cdot cos\,\theta_v \\ y_v + l_{GC} \cdot sin\,\theta_v \end{bmatrix}.$$

The values for x_{rows_init} and y_{rows_init} are stored for each particle, however we have not explicitly included them as part of the state vector, since they are not quantities that we are attempting to estimate. Instead they can be thought of as parameters that form part of the measurement equation for the crop row tracking.

The offset τ_{rows} is the length of the line projected from point C perpendicular to the rows that intersects a line parallel to the rows passing through point I. Our measurement equation for the crop rows is therefore

$$\mathbf{z}_{rows} = \begin{bmatrix} \tau_{rows} \\ \emptyset_{rows} \end{bmatrix}$$

$$= \begin{bmatrix} (x_C - x_{rows\_init})\, sin(\beta) + (y_C - y_{rows\_init})\, cos(\beta) \\ \theta_v - \beta \end{bmatrix}$$

where x_C and y_C are the coordinates of the visual navigation point C is given by

$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = \begin{bmatrix} x_v + l_{GC}\, cos\, \theta_v \\ x_v + l_{GC}\, sin\, \theta_v \end{bmatrix}.$$

This measurement equation assumes the crop rows are straight and aligned well with the nominal crop direction.

The update step (re-weighting of the particles) is performed both when a GPS message is received, and when a row tracking observation is received.

5 Experimental Setup

The experiments were performed on a broad acre sorghum stubble field located near Emerald in Australia. The sorghum crop was planted in the previous season with an RTK-GPS guided tractor with all rows straight and parallel except for rows around the headland turning areas at the ends of the field and around obstacles such as a power pole.

We conducted the first experiment live with the robot's position estimated by the filter using the Skytraq GPS receiver with RTK corrections applied, wheel odometry and IMU data. The robot simulated spraying a 6 hectare area of the field by guiding itself over the field in a series of parallel swaths (i.e. "lawnmower" pattern) between waypoints at each end of the field. The direction of travel is aligned with the direction of the crop rows. The robot has a 5 metre wide spray boom implement, which due to the nozzles, sprays approximately an extra 0.5 metres on either side. In this experiment there were no sensor dropouts.

In the second experiment we test the ability of our positioning system, including the vision based row tracking, to handle dropouts of the RTK correction signal. RTK dropouts could be caused by 3G wireless signal loss, CORS network failure, or telecommunications network failure. To conduct the experiment the sensors data is replayed from a detailed log file and the filter is re-run. The correction data dropout is simulated by using the uBlox GPS that had no correction signal.

In the third experiment we test the ability of our positioning system to handle GPS dropouts while using our vision based row tracking algorithm. We examine the degradation in accuracy that occurs when the GPS signal is lost entirely, by supplying the filter with an intermittent GPS signal with dropouts of various lengths. The filter is re-run using the log data and supplying the filter with 10 seconds of GPS data, before removing GPS input for 10 seconds, 20 seconds or 60 seconds.

When performing an area coverage task such as spraying weeds, the implement (i.e. spray boom) should cover the entire field exactly once, while minimising missing sections and overlap. Since the distance travelled while adjacent to the field boundaries is usually small compared to the total distance travelled, the most desirable quantity to optimise for coverage efficiency is the pass-to-pass error. For this reason we use the pass-to-pass error in position to compare the performance of different filter combinations. Pass-to-pass distance is defined here as the distance to the previous pass in a direction perpendicular to the direction of the rows. We manually exclude sections where the vehicle is turning around at the ends of the field or when avoiding obstacles as the vehicle is not travelling parallel to the rows. We use the Novatel GPS/INS as a ground truth for the results.

6 Results

6.1 Live Coverage

The path taken by the robot while being guided by our filter (as recorded by the ground truth reference system) is shown in Figure 5. With the robot performing 5 m wide swaths travelling at 5 km/hr it covered the field in approximately 2 hours covering a distance of around 10 km. The three major diversions from this path were due to the robot avoiding obstacles. Figure 6 shows the pass-to-pass error for the filter output. The RMS pass-to-pass error in this experiment is 0.18 m while the 95th percentile error is 0.28m. These errors are is less than the 0.5 m wide side-spray. Assuming a 6 m wide spray, the robot missed 2.6% of the area and covered an extra 9.7% of the area. The GPS and IMU sensors used for these results only cost hundreds of dollars, and yet were able to provide useful accuracy for the coverage task.

Accuracies were not consistently in the region of 2cm that are achievable by RTK-GPS, which can partly be attributed to not waiting while static for an unambiguous RTK fix before beginning the experiment. The alternating nature of the error in Figure 6 is due to an error in the code causing a small constant bias in the IMU yaw rate which caused the vehicle to always be slightly to the left of its desired path. This error was corrected for the post-processed results presented in the following sections, which further improved the accuracy of the filter (see Figure 7).

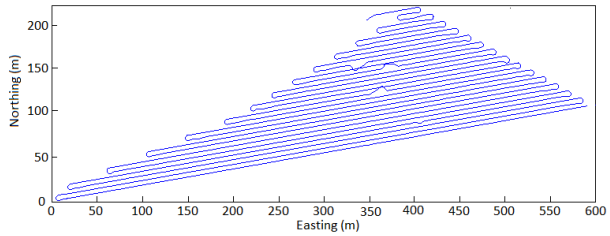

Figure 5: Path taken by the robot according to the Novatel reference system in a local reference frame. The three divergences from the "lawnmower pattern" are from robot avoiding obstacles as part of a separate study.

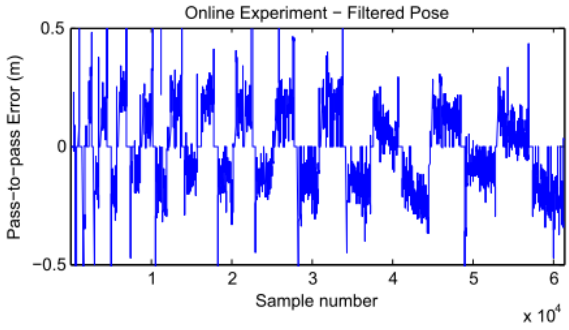

Figure 6: Pass-to-pass error for the filter output used to guide the robot during the experiment.

6.2 RTK Corrections Dropout

The pass-to-pass error plots for running the filter with RTK GPS with and without row tracking are shown in Figure 7. Pass-to-pass error plots for running the filter with only GPS both with and without row tracking are shown in Figure 8. Table 1 shows the RMS and 95th percentile error for these four cases. The addition of row tracking into the filter made virtually no difference to the result for RTK GPS, which is unsurprising since the RTK GPS is already providing very good accuracy. However, for the case of no GPS corrections, row tracking improved coverage efficiency, with RMS error reduced by 28% and 95th percentile error by 42%. These results indicate that row tracking is a useful tool for improving accuracy during periods of time when RTK correction information is lost (i.e. network outages).

Figures 9 and 10 each show a section of the path output by the filter compared to the ground truth and the raw GPS position. For the case of RTK GPS (Figure 9), the filter output follows the GPS input almost exactly. For uncorrected GPS (Figure 10), the filtered position generally follows the shape of the ground truth, but is offset. The uncorrected GPS is quite noisy. Figure 10 demonstrates how noisy the uncorrected GPS data is, and shows that filtering does a good job at removing this noise, with the output generally the same shape as the ground truth path, however it is offset by a slowly varying amount.

Filter sensor combination	RMS Error(m)	95% Error (m)
Skytraq (RTK GPS)	0.07	0.11
Skytraq (RTK GPS) + row tracking	0.08	0.11
uBlox (GPS)	0.73	1.05
uBlox (GPS) + row tracking	0.52	0.60

Table 1: Pass-to-pass error results.

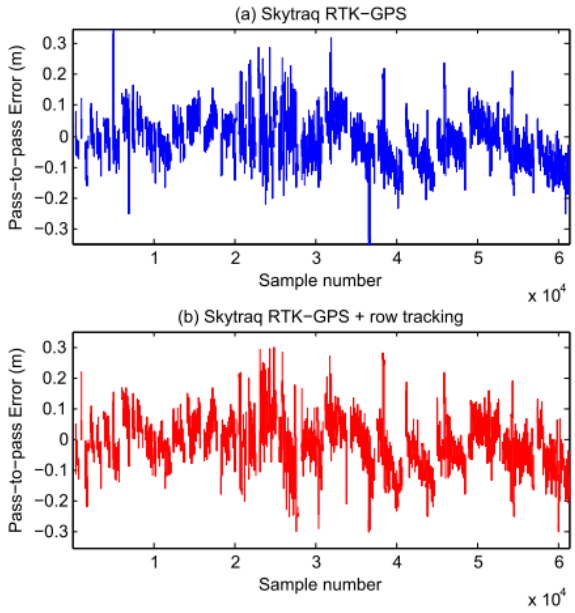

Figure 7: Pass-to-pass error of filter output for the Skytraq GPS (with RTK corrections) without row tracking (top) and with row tracking (bottom). Our row tracking system doesn't improve the accuracy when using RTK GPS.

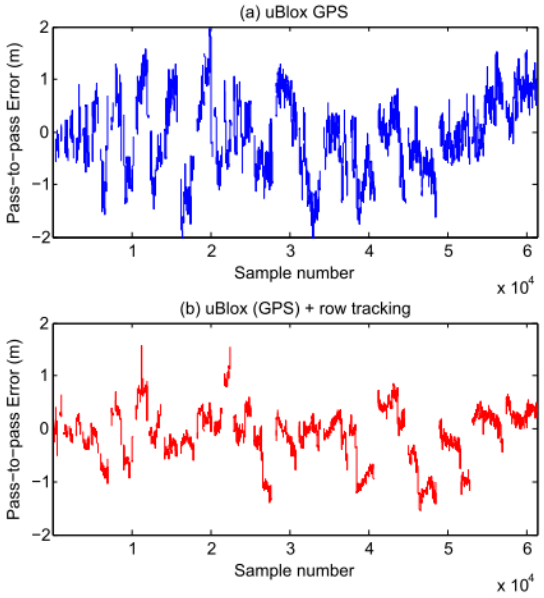

Figure 8 Pass-to-pass error of filter output for the uBox GPS (no RTK correction) without row tracking (top) and with row tracking (bottom). Pass-to-pass error improves using row tracking.

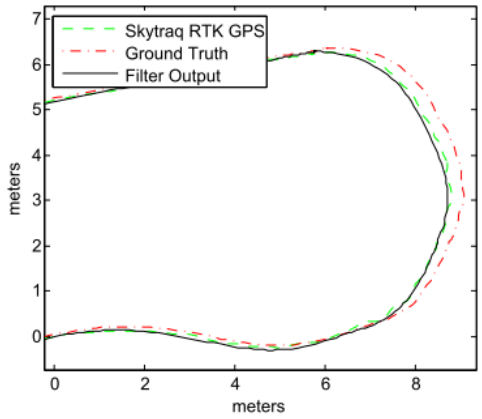

Figure 9: Example path output by the filter using the Skytraq GPS and row tracking (black) matches the Skytraq GPS (green) quite closely.

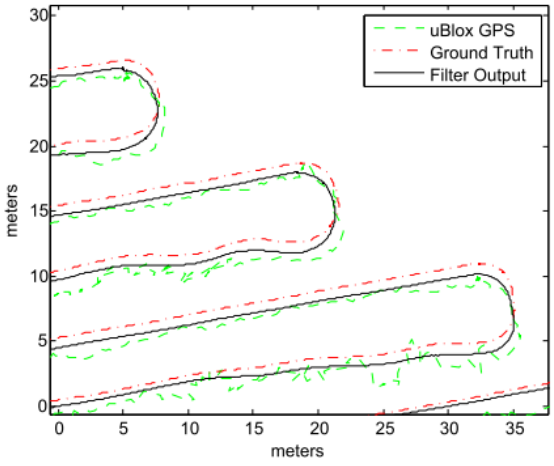

Figure 10 Filter output for uncorrected GPS and row tracking (black) is smooth and matches the ground truth, but with a slowly varying offset. The unfiltered GPS data (green) is quite noisy.

6.3 GPS Outages

Figure 12 shows the effect of suppling the filter with 10 seconds of RTK GPS data, before removing GPS input for 10 seconds, 20 seconds or 60 seconds. Table 2 and Figure 11 show the RMS and 95th percentile error for each of the situations tested. The filter accuracy drops as the length of the GPS outages increases, and for the case of 60 second outages the filter solution diverged after 30 minutes into the approximately 2 hour test. This result indicates that for many applications, field operations may be able to continue for some time without GPS at all. In practice, position uncertainty can be measured by the spread of the particles in the filter and the vehicle can be stopped when accuracy becomes unacceptable.

The largest errors occurred while turning around at the ends of the rows without GPS. This is expected since row tracking is unavailable at the ends of the field and so the particle filter is using odometry information alone.

GPS Denial Time (seconds) (10 seconds on, x seconds off)	RMS Error(m)	95% Error (m)
0	0.08	0.15
10	0.10	0.22
20	0.23	0.39
60	0.52	1.1

Table 2: Pass-to-pass error of filter output under various GPS outage lengths in-between 10 second periods of GPS availability.

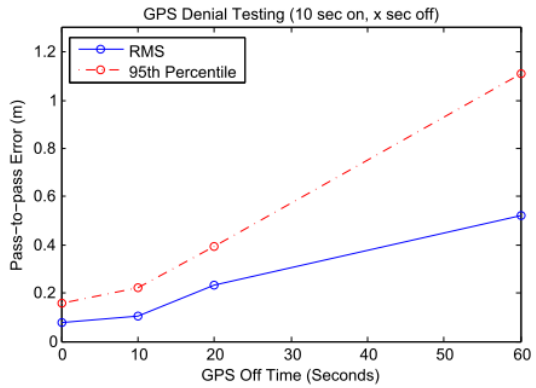

Figure 11: RMS and 95th percentile pass-to-pass error for various GPS denial times.

7 Conclusion

This paper demonstrated the performance of a positioning system that is suitable for coverage tasks by low cost agricultural robots. The live results show that the positioning system is able to successfully guide a robot to cover a large area using a free library to integrate corrections. When combined with our new vision based row tracking system, we demonstrated that the positioning system is able to handle long correction signal dropouts and regular intermittent total GPS dropouts. This means that the robot would be able to continue with its coverage task despite these problems. This approach of combining multiple inexpensive sensors with local row tracking shows promise as a practical localisation system for low-cost agricultural robots to automate tasks currently performed on farms by large manned machinery.

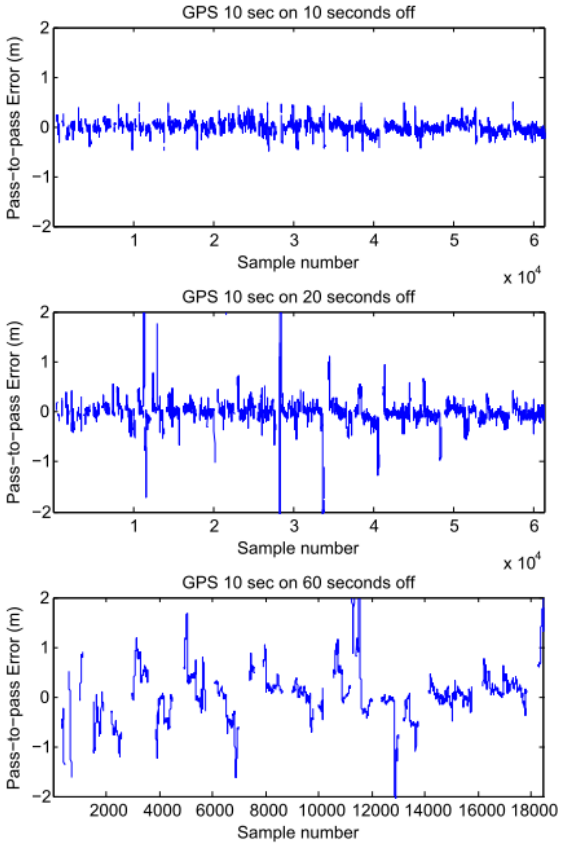

Figure 12: Pass-to-pass errors for various intermitent GPS outage lengths in-between 10 second periods of GPS availability.

Acknowledgements

This work was supported in part by the Australian Research Council Linkage Project LP110200375 "Robotics for Zero Tillage Agriculture" awarded to the Queensland University of Technology, Swarm Farm Robotics and the Australian Centre for Field Robotics.

References

[Bar-Shalom et al., 2004] Bar-Shalom, Yaakov, X Rong Li, and Thiagalingam Kirubarajan. 2004. Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software. John Wiley & Sons.

[Biber, 2010] Ruckelshausen, A., P. Biber, M. Dorna, H. Gremmes, R. Klose, A. Linz, F. Rahe et al. "BoniRob–an autonomous field robot platform for individual plant phenotyping." Precision agriculture 9 (2009): 841.

[Jensen et al., 2012] Jensen, Morten Larsen, Tom Simonsen, Rasmus N. Jørgensen. 2012. "Evaluating the performance of a low-cost GPS in precision agriculture applications". First RHEA International Conference on Robotics and associated High-technologies and Equipment for Agriculture.

[Kise et al., 2008] Kise, Michio, and Qin Zhang. 2008. "Development of a Stereovision Sensing System for 3D Crop Row Structure Mapping and Tractor Guidance." Biosystems Engineering 101 (2) (October): 191–198.

[Li and Wang, 2012] Li, T, and J Wang. 2012. “Some Remarks on GNSS Integer Ambiguity Validation Methods.” Survey Review 44 (326) (July): 230–238.

[Mao et. al., 1999] Mao, Ailin, CGA Harrison, and TH Dixon. 1999. “Noise in GPS Coordinate Time Series.” Journal of Geophysical Research.

[Osterman et al., 2013] Osterman, A, T Godeša, and M Hočevar. 2013. “Introducing Low-cost Precision GPS/GNSS to Agriculture.” In Actual Tasks on Agricultural Engineering. Proceedings of the 41. International Symposium on Agricultural Engineering, Opatija, Croatia, 19-22 February 2013., 229–239. University of Zagreb Faculty of Agriculture.

[Søgaard and Olsen, 2003] Søgaard, H.T., and H.J. Olsen. 2003. “Determination of Crop Rows by Image Analysis Without Segmentation.” Computers and Electronics in Agriculture.

[Stempfhuber and Buchholz, 2001] Stempfhuber, W, and M Buchholz. 2011. “A PRECISE , LOW-COST RTK GNSS SYSTEM FOR UAV APPLICATIONS” Remote Sensing and Spatial Information Sciences , XXXVIII (September): 14–16.

[Taki et al., 2011] Takai, Ryosuke, Noboru Noguchi, Liangliang Yang, and Ze Zhang. 2011. “Crawler-Type Robot Tractor Using Multi-GNSS with QZSS

[Takasu and Yasuda, 2009] Takasu, Tomoji, and Akio Yasuda. 2009. “Development of the Low-cost RTK-GPS Receiver with an Open Source Program Package RTKLIB.” International Symposium on GPS/GNSS.

[Thurrowgood et al., 2009] Thurrowgood, Soccol, Moore, Bland, and Srinivasan. 2009. “A vision based system for attitude estimation of UAVs”. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 2009, pp. 5725–5730.

[Tillet and Hague, 2002] Tillett, N.D., T. Hague, and S.J. Miles. 2002. “Inter-row Vision Guidance for Mechanical Weed Control in Sugar Beet.” *Computers and Electronics in Agriculture* 33 (3) (March): 163–177.